

Робот сервиса Wiki

Прогнозирование спроса на аренду велосипедов в системе велопроката

Обновлено 22 мая 2025, 18:23

Содержание страницы

Описание проекта

[Бизнес-обоснование](#)[Проблема](#)[Цели проекта](#)[Целевая аудитория](#)[Ожидаемые выгоды](#)[Риски и ограничения](#)[Функциональные требования](#)[Основные функции](#)[Потоки данных](#)[Нефункциональные требования](#)[Ограничения](#)[Архитектура системы](#)[Компоненты](#)[Хранение моделей](#)

Описание проекта

Проект направлен на разработку интеллектуальной прогностической системы для предсказания спроса на аренду велосипедов в системе велопроката.

Система использует датасет UCI Bike Sharing Dataset (почасовые данные за 2011–2012 годы) для обучения моделей и симуляции работы велопроката.

Основная цель — обеспечить точные прогнозы спроса (на ближайший час), улучшить клиентский опыт и создать прототип для курсовой работы, демонстрирующий навыки проектирования интеллектуальных систем.

Бизнес-обоснование

Проблема

Операторы велопроката сталкиваются с трудностями в управлении парком велосипедов из-за непредсказуемого спроса, что приводит к следующим проблемам:

- Недостаток велосипедов в часы пик, что снижает удовлетворённость клиентов.
- Избыток велосипедов в непопулярное время, увеличивая затраты на обслуживание.
- Отсутствие персонализированных прогнозов для крупных клиентов (например, транспортных компаний), что ограничивает возможности планирования.

Цели проекта

1. **Улучшение клиентского опыта:** Обеспечить доступность велосипедов в нужное время, повышая лояльность пользователей.
2. **Монетизация возможностей:** Реализовать платную подписку с индивидуальными моделями прогнозирования для премиум-пользователей.
3. **Академическая ценность:** Создать прототип для курсовой работы, демонстрирующий навыки проектирования микросервисной архитектуры и интеллектуальных систем.

Целевая аудитория

- **Операторы велопроката:** Для планирования логистики и управления парком.
- **Частные пользователи:** Горожане, планирующие поездки на велосипедах.
- **Студент (автор курсовой):** Для демонстрации навыков проектирования.

Ожидаемые выгоды

Выгода	Описание	Кому

Экономия ресурсов	Снижение затрат на перераспределение велосипедов за счёт точных прогнозов.	Операторы велопроката
Повышение дохода	Платная подписка с индивидуальными моделями привлечёт клиентов.	Разработчик системы
Улучшение UX	Доступность велосипедов повышает удовлетворённость пользователей.	Частные пользователи
Академическая ценность	Прототип демонстрирует навыки проектирования микросервисной архитектуры.	Студент (автор курсовой)

Риски и ограничения

Риск	Описание	Митигация
Недостаток фактических данных	Отсутствие реальных данных велопроката в реальном времени.	Симуляция данных на основе UCI Bike Sharing Dataset.
Нагрузка на систему	Высокая частота дообучения увеличивает нагрузку на PostgreSQL и Redis.	Ограничение частоты дообучения и кэширование моделей.
Ограничение времени	Сложность реализации всех микросервисов в рамках курсовой.	Упрощение прототипа: 1 платный пользователь, локальное хранение.

Функциональные требования

Основные функции

ID	Функция	Описание	Приоритет
FR-01	Прогноз спроса	Прогнозирование спроса на аренду велосипедов на следующий час.	Высокий
FR-02	Бесплатный режим	Прогноз с общей моделью (кэшируется в In-Memory Cache).	Высокий
FR-03	Платный режим	Прогноз с индивидуальной моделью (кэшируется в Redis).	Высокий
FR-04	Обновление фактических данных	Сохранение фактических данных (actual_rentals) в PostgreSQL.	Средний
FR-05	Дообучение моделей	Инкрементное дообучение индивидуальных моделей (раз в неделю).	Средний
FR-06	Управление пользователями	Регистрация пользователей и управление подписками (free/premium).	Средний
FR-07	История прогнозов	Хранение истории прогнозов и фактических данных для анализа точности.	Низкий

Потоки данных

1. Прогноз:

- Клиент → API Gateway → Service Prepare (предобработка) → Kafka (preprocessed-data-free/premium) → Service Predict → Kafka (prediction-results) → API Gateway → Клиент.

2. Обновление фактических данных:

- Клиент → API Gateway → Service Update Rentals → PostgreSQL → Kafka (actual-data-updated).

3. Дообучение:

- Service Update Models → PostgreSQL (чтение новых данных) → дообучение модели → Redis (очистка кэша) → Kafka (model-retrained).

4. Логирование:

- Все компоненты → Grafana/Loki (структурированные JSON логи).

Нефункциональные требования

ID	Категория	Требование	Описание
NFR-01	Производительность	Время обработки запроса	Менее 1 секунды для 95% запросов (с учётом кэширования).

NFR-02	Масштабируемость	Поддержка пользователей	До 100 пользователей для прототипа (1 платный пользователь в курсовой).
NFR-03	Доступность	Время работы	99.9% uptime (кроме планового обслуживания).
NFR-04	Надёжность	Обработка ошибок	Логирование всех ошибок в Grafana/Loki, автоматическое уведомление.
NFR-05	Безопасность	Аутентификация	Простая аутентификация для платных пользователей (например, API токены).
NFR-06	Хранение данных	Объём базы данных	До 1 ГБ для прототипа (PostgreSQL).
NFR-07	Мониторинг	Логирование	Визуализация метрик (время обработки, точность прогнозов) в Grafana.
NFR-08	Совместимость	Технологии	Python (FastAPI, XGBoost), PostgreSQL, Kafka, Redis, Grafana/Loki.

Ограничения

ID	Ограничение	Описание	Влияние
C-01	Ограничение времени	Курсовая работа должна быть завершена за 1–2 месяца.	Упрощение прототипа: 1 платный пользователь, локальное хранение.
C-02	Отсутствие реальных данных	Нет доступа к API велопроката для фактических данных.	Симуляция на основе UCI Bike Sharing Dataset.
C-03	Ограничение ресурсов	Ограниченный доступ к вычислительным ресурсам.	Ограничение числа индивидуальных моделей и дообучения.
C-04	Не нейросетевые модели	Используются XGBoost для простоты реализации.	Ограничение точности (без инкрементального обучения).

Архитектура системы

Компоненты

1. **Клиентское приложение:** React для ввода данных и отображения прогнозов.
2. **API Gateway:** FastAPI/Nginx для маршрутизации и балансировки.
3. **Service Prepare:** Микросервис предобработки данных (нормализация, кодирование) → отправка в Kafka.
4. **Service Predict:** Выполняет прогнозы с общей моделью (In-Memory Cache) или индивидуальной (Redis/PostgreSQL).
5. **Service Update Rentals:** Обновляет фактические данные в PostgreSQL.
6. **Service Update Models:** Дообучает индивидуальные модели (раз в неделю) и обновляет общую модель (вручную).
7. **Kafka:** Брокер сообщений для асинхронной передачи данных (топики: preprocessed-data-free/premium, prediction-results, actual-data-updated, model-retrained).
8. **PostgreSQL:** Хранит данные, прогнозы, модели (индивидуальные как BYTEA).
9. **Redis:** Кэширует индивидуальные модели с TTL (24 часа).
10. **Grafana/Loki:** Мониторинг и визуализация логов.

Хранение моделей

- **PostgreSQL:** Таблица models, поле model_data как BYTEA.
- **Redis:** Кэширование индивидуальных моделей (TTL 24 часа).
- **In-Memory Cache:** Общая модель хранится в памяти сервиса прогнозирования.

Ответственный: Дуванов Илья Алексеевич

Срок: 23.05.2025

Контакты:

duvanovilja@gmail.com

t.me/dyvik